DERWENT-ACC-NO: 1995-102787
DERWENT-WEEK: 199514

TITLE: Software renewal method for communications network - updating and classifying software independent of physical memory configuration from remote location

PATENT-ASSIGNEE: NIPPON TELEGRAPH & TELEPHONE CORP[NITE]

PRIORITY-DATA: 1993JP-0191842 (July 7, 1993)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
JP 07028724 A	January 31, 1995	N/A	008	G06F 013/00

APPLICATION-DATA:

PUB-NO	APPL-DESCRIPTOR	APPL-NO	APPL-DATE
JP 07028724A	N/A	1993JP-0191842	July 7, 1993

INT-CL (IPC): G06F009/06; G06F013/00 ; H04M003/00 ; H04Q003/545

ABSTRACTED-PUB-NO: JP 07028724A

BASIC-ABSTRACT: The software renewal method involves using an operating system to remotely renew software through a communication network. An operation system controls the operation which carries out updating of the network elements. Each network element notifies the updating situation of the software renewal by state change to an operation system. The attribute expresses the cut return state on the gamma state from a low-speed memory media and cut return impossible state and a file memory which can be cut returned and a hard disk.

USE/ADVANTAGE - File renewal, location data change of software of network elements in communication network from remote location. Controls operation interface and control sequence becomes logical. Facilitates inter-operability between different hardware and software.

CHOSEN-DRAWING: Dwg.1/18

TITLE-TERMS:

SOFTWARE RENEW METHOD COMMUNICATE NETWORK UPDATE CLASSIFY SOFTWARE INDEPENDENT
PHYSICAL MEMORY CONFIGURATION REMOTE LOCATE

DERWENT-CLASS: T01 W01

EPI-CODES: T01-F05; T01-H07C; W01-B02A1; W01-C02A7;

SECONDARY-ACC-NO:
Non-CPI Secondary Accession Numbers: N1995-080898

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（A）**　(11)特許出願公開番号

特開平7－28724

(43)公開日　平成７年(1995)１月31日

(51)Int.Cl.[6]		識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ	13/00	３５１　Ｈ	7368－5B		
	9/06	５４０　Ａ	9367－5B		
Ｈ０４Ｍ	3/00	Ｅ	8426－5K		
Ｈ０４Ｑ	3/545		8843－5K		

審査請求　未請求　請求項の数２　ＦＤ　（全８頁）

(21)出願番号　特願平5－191842

(22)出願日　平成５年(1993)７月７日

(71)出願人　000004226
日本電信電話株式会社
東京都千代田区内幸町一丁目１番６号

(72)発明者　高橋　和秀
東京都千代田区内幸町一丁目１番６号日本電信電話株式会社内

(72)発明者　老松　敏雄
東京都千代田区内幸町一丁目１番６号日本電信電話株式会社内

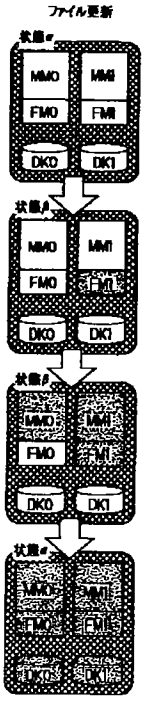
(74)代理人　弁理士　山本　恵一

(54)【発明の名称】　ソフトウェア更新制御方法

(57)【要約】

【目的】　本発明はＯｐＳとＮＥの独立な発展を可能として相互接続性を保証し、ＮＥ側の更新種別の変更あるいは追加、新機種のＮＥの導入に対して柔軟に対応するソフトウェア更新制御方法を提供することを目的とする。

【構成】　本発明は上記目的を達成するために、ソフトウェア更新の更新状況を、オペレーションシステムのプログラムのクラスオブジェクト内の更新状態を表す属性の値により表現する。また切戻し不可能状態である状態α、ファイルメモリ等の高速記憶媒体から切戻し可能状態である状態β及びハードディスク等の低速記憶媒体から切戻し可能状態である状態γで表現する。よって、ＮＥに対するＯｐＳの制御操作インタフェースをＮＥの機種、更新種別及びベンダに非依存となるために、制御操作インタフェース及び制御操作シーケンスの論理化により、ＯｐＳとＮＥの独立な発展を可能とし、相互接続性を保証できる。



1

【特許請求の範囲】

【請求項１】　オペレーションシステムが交換機等のネットワークエレメントを保守するために、オペレーションシステムが遠隔から通信網を介してネットワークエレメントのソフトウェア更新を実施するためにネットワークエレメントを制御するソフトウェア更新制御方法において、

ソフトウェア更新の更新状況を、オペレーションシステムのプログラムのクラスオブジェクト内の更新状態を表す属性の値により表現し、かつ切戻し不可能状態である状態α、ファイルメモリ等の高速記憶媒体から切戻し可能状態である状態β及びハードディスク等の低速記憶媒体から切戻し可能状態である状態γで表現し、

オペレーションシステムがネットワークエレメントに対して行う制御操作を実行β、実行γ、バックアップα、バックアップγ及び切戻しにより表現し、

オペレーションシステムはソフトウェア更新の起動契機によりクラスオブジェクトのインスタンスを生成し、

ネットワークエレメントがオペレーションシステムに対してソフトウェア更新の更新状況を状態変更により通知し、

オペレーションシステムはネットワークエレメントからの通知を契機にクラスオブジェクトのインスタンスの属性値を変更することにより、ソフトウェア更新の更新状況を管理し、インスタンスの状態遷移によりソフトウェア更新を制御することを特徴とするソフトウェア更新制御方法。

【請求項２】　他のネットワークエレメントの導入や他のソフトウェア更新種別の追加に対しては前記クラスオブジェクトのサブクラス化により状態を表す属性を追加する請求項１記載のフトウェア更新制御方法。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明はソフトウェア更新制御方法に関し、特に交換機等のネットワークエレメント（以下ＮＥと略す）の保守運用を行うシステムであるオペレーションシステム（以下ＯｐＳと略す）がＮＥを保守する環境においてＯｐＳが遠隔から通信網を介してＮＥのソフトウェア更新（ファイル更新、パッチ投入、所データ変更等）を実施する場合、ＯｐＳがＮＥを制御する方法に関する。

【０００２】

【従来の技術】現在、図１３及び図１４に示すような様々な物理メモリ構成を持ったＮＥが存在する。図１３はメインメモリ（以下ＭＭと略す）、ファイルメモリ（以下ＦＭと略す）及びハードディスク（以下ＤＫと略す）の二重化メモリ構成を持つＮＥの例を示す図である。また図１４はＭＭ及びＤＫの二重化メモリ構成を持つＮＥの例を示す図である。

【０００３】ＯｐＳがＮＥに対して行うソフトウェア更新の業務には、交換機のソフトウェアバグの修正や機能追加を目的としてシステムファイル全体を入れ替えるファイル更新と、交換機のソフトウェアバグの修正や機能追加を目的としてシステムファイルを部分的に修正するために書き替えるパッチ投入と、交換機の装置増設、回線増設、局番変更等に伴って所データを部分的に変更する所データ変更等の業務がある。これらの業務シーケンスの中で、ＯｐＳはＮＥに対して以下のような様々な制御操作を行う。

2

【０００４】図１５は図１３の物理メモリ構成を持つＮＥのソフトウェア更新シーケンスのうちファイル更新を示し、図１６はパッチ投入を示し、図１７は所データ変更を示す。また図１８は図１４の物理メモリ構成を持つＮＥのソフトウェア更新シーケンスのうちファイル更新を示す。

【０００５】まず図１５に示す図１３の物理メモリ構成を持つＮＥのファイル更新シーケンスについて説明すると、最初のFUP NEW/の制御操作により１系のＦＭ１に新ファイルがロードされる（この処理は図中編掛で示す）。次のCNV FUP OK/ の制御操作によりファイル更新再開処理が行われ、両系のＭＭ０とＭＭ１に新ファイルがロードされ、ＮＥは１系で立ち上がる。最後のFUP FIN/の制御操作により０系のＦＭ０に新ファイルがコピーされ、引き続いて両系のＤＫ０とＤＫ１にバックアップファイルが収集される。

【０００６】次に図１６に示す図１３の物理メモリ構成を持つＮＥのパッチ投入シーケンスについて説明すると、最初のMLD PAT,PUP/の制御操作により新パッチが両系のＭＭ０とＭＭ１に投入される。次のMLD PAT,SUP/の制御操作により新パッチが両系のＦＭ０とＦＭ１に新ファイルに投入される。最後のFST BFM,LAF/によりが両系のＤＫ０とＤＫ１にバックアップファイルが収集される。

【０００７】図１７に示す図１３の物理メモリ構成を持つＮＥの所データ変更シーケンスについて説明すると、最初のTCA/の制御操作により両系のＭＭ０とＭＭ１の中間テーブル（図中太い罫線で示す）の所データが変更される。次のTCH NEW/の制御操作により両系のＭＭ０とＭＭ１の中間テーブルが変更される。ODB/の制御操作により両系のＦＭ０とＦＭ１が変更され、最後のFST BFM,LAF/により両系のＤＫ０とＤＫ１にバックアップファイルが収集される。

【０００８】図１８に示す図１４の物理メモリ構成を持つＮＥのファイル更新シーケンスについて説明すると、最初のFUP NEW/の制御操作により１系のＭＭ１に新ファイルがロードされる。次のCNV FUP OK/ の制御操作によりファイル更新再開処理が行われ、ＮＥは１系で立ち上がる。最後のFUP FIN/の制御操作により両系のＤＫ０とＤＫ１にバックアップファイルが収集される。

【０００９】

3

【発明が解決しようとする課題】通常、従来のソフトウェア更新の制御方法では、更新種別に応じて、ＯｐＳがＮＥに対して行う制御操作インタフェース及び制御シーケンスは異なるため、従来例によれば、ＮＥの仕様変更に伴い更新種別が変更あるいは追加される場合、ＯｐＳの制御シーケンスを変更あるいは追加しなければならない。従って、ＯｐＳ側のプログラムはＮＥ側のプログラムの変更の影響を受けるため、ＮＥとＯｐＳの独立な発展ができないという問題点がある。

【００１０】またＮＥの物理メモリの構成に応じて、ＯｐＳがＮＥに対して行う制御操作インタフェース及び制御シーケンスが異なるため、ＯｐＳが保守するＮＥの機種（例えば交換機の場合Ｄ７０，ＩＳＭ，Ｄ５１，ＤＭＳ１０など）が増える場合、従来例によれば、ＯｐＳの制御操作インタフェース及び制御操作シーケンスを追加しなければならない。従って、今後通信ネットワークの高度化にともない様々なベンダ（特に海外northern telecom, ericsson, ATT など）がＮＥを提供するマルチベンダ環境を考えるとＮＥとＯｐＳの相互接続性が保証されないという問題点がある。

【００１１】本発明の目的は、制御操作インタフェース及び制御操作シーケンスの論理化により、ＯｐＳとＮＥの独立な発展を可能とし、相互接続性を保証することにある。従って、本発明におけるＯｐＳはＮＥ側の更新種別の変更あるいは追加、新機種のＮＥの導入に対して柔軟に対応することができる。

【００１２】

【課題を解決するための手段及び作用】本発明は前記問題点を解決するために、ソフトウェア更新の更新状況を、ＯｐＳのプログラムのクラスオブジェクト内の更新状態を表す属性の値により表現し、かつ切戻し不可能状態である状態α、ファイルメモリ等の高速記憶媒体から切戻し可能状態である状態β及びハードディスク等の低速記憶媒体から切戻し可能状態である状態γで表現する。また、ＯｐＳがＮＥに対して行う制御操作を実行β、実行γ、バックアップα、バックアップγ及び切戻しにより表現する。さらに、ＯｐＳはソフトウェア更新の起動契機によりクラスオブジェクトのインスタンスを生成し、ＮＥがＯｐＳに対してソフトウェア更新の更新状況を状態変更により通知する。そして、ＯｐＳはＮＥからの通知を契機にクラスオブジェクトのインスタンスの属性値を変更することにより、ソフトウェア更新の更新状況を管理し、インスタンスの状態遷移によりソフトウェア更新を制御する。

【００１３】また、他のＮＥの導入や他のソフトウェア更新種別の追加に対してはクラスオブジェクトのサブクラス化により状態を表す属性を追加する。従って、本発明は前記問題点を解決でき、ＮＥに対するＯｐＳの制御操作インタフェースをＮＥの機種、更新種別及びベンダに非依存となるために、制御操作インタフェース及び制御操作シーケンスの論理化により、ＯｐＳとＮＥの独立な発展を可能とし、相互接続性を保証できる。

4

【００１４】

【実施例】以下、本発明の実施例について図面に基づいて説明する。はじめに、ソフトウェア更新の途中経過を、切戻し不可能状態（この状態を状態αと称す）、ファイルメモリ等の高速記憶媒体から切戻し可能状態（この状態を状態βと称す）及びハードディスク等の低速記憶媒体から切戻し可能状態（この状態を状態γと称す）の３状態により表現し、制御操作インタフェースを実行β、実行γ、バックアップα、バックアップγ及び切戻しにより表現する。

【００１５】図１は本実施例における、図１３の物理メモリ構成を持つＮＥのソフトウェア更新シーケンスのうちファイル更新シーケンスを示し、図２は本実施例における所データ変更シーケンスを示し、図３は本実施例におけるパッチ投入シーケンスを示す。また図４は本実施例における、図１４の物理メモリ構成を持つＮＥのソフトウェア更新シーケンスのうちファイル更新シーケンスを示す。なお、各シーケンスの説明は後述するものとする。

【００１６】ここで、図５はソフトウェア更新クラスの継承関係を示す図であり、更新状態及び制御操作を拡張（詳細化）したい場合このクラスをサブクラス化してサブクラスで新しい更新状態及び制御操作を定義する。例えば「状態β」を「状態β１」と「状態β２」に拡張（詳細化）したい場合、新しいクラスの「更新クラス１」を定義する。このクラスは「状態β」を拡張した属性として「β更新状態」を持ち、その値として「状態β１」及び「状態β２」をとる。図６～８にはクラスオブジェクトの継承関係を示す。図６～８及びその状態遷移規則を示す下記の表１に示すように、属性として更新状態（値：α、β、γ）、交換機に対するＯｐＳの制御操作（インタフェース）として実行β、実行γ、バックアップα、バックアップγ及び切戻しを持つソフトウェア更新クラスをスーパクラスとして定義する。また、２つある状態βを区別するために、属性として更新状態β（値：β１、β２）、制御操作として実行β２を持つ更新クラス１を、状態γを区別するために、属性として更新状態γ（値：γ１、γ２）、制御操作として実行γ２を持つ更新クラス２をソフトウェア更新クラスのサブクラスとして定義する。

【００１７】

【表１】

5　　6

操作前		操作	操作後	
現状態	ファイル種別		次状態	ファイル種別
状態α	旧ファイル	実行β	状態β	新ファイル
状態β	新ファイル	バックアップγ	状態β	新ファイル
状態α	旧ファイル	実行γ	状態γ	新ファイル
状態γ	新ファイル	バックアップα	状態α	新ファイル
状態β	新ファイル	切戻し	状態α	旧ファイル
状態γ	新ファイル	切戻し	状態α	旧ファイル

【００１８】ＯｐＳのプログラムに、これらのクラスオブジェクト及び状態遷移規則を配備し、ＮＥのプログラムに状態遷移規則を配備する。

【００１９】保守者等が、ＯｐＳに対して、目的のソフトウェア更新を指示すると、ＯｐＳは対応するインスタンスを生成する。初期状態は更新状態がα状態であるときとする。ＯｐＳが、ＮＥに対して対応する制御操作を行うと、ＮＥは対応するメモリの更新を行い、状態遷移規則から次に状態を決定して状態遷移する。次に、ＮＥが状態遷移した旨をＯｐＳに対して通知すると、ＯｐＳはインスタンスの更新状態を変更し、状態遷移規則から次の制御操作を決定して、ＮＥに対して制御操作を行う。目的のソフトウェア更新が終了し、インスタンスの更新状態が再びα状態となるまで、以下同様に繰り返す。

【００２０】図９は、本実施例における、図１３に示す物理メモリ構成を持つＮＥについてのファイル更新シーケンスを示す図である。同図は図１に対応するものであり、同図において、最初の実行βの制御操作により、１系のＦＭ１に新ファイルがロードされ、インスタンスは状態β（β１）に遷移する。次の実行β２の制御操作により、ファイル更新再開処理が行われ、両系のＭＭ０，ＭＭ１に新ファイルがロードされ、ＮＥは１系で立ち上がり、インスタンスは状態β２に遷移する。最後の実行αの制御操作により、０系のＦＭ０に新ファイルがコピーされ、引き続いて両系のＤＫ０，ＤＫ１に前日ファイル及び保証ファイルが収集され、インスタンスは状態αに遷移する。

【００２１】図１０は、本実施例における、図１３に示す物理メモリ構成を持つＮＥについてのパッチ投入シーケンスを示す図である。同図は図３に対応するものであり、同図において、最初の実行γの制御操作により、新パッチが両系のＭＭ０，ＭＭ１に投入され、インスタンスは状態γ（γ１）に遷移する。次の実行γ２の制御操作により新パッチが両系のＦＭ０，ＦＭ１に投入され、インスタンスは状態γ２に遷移する。最後の実行αにより両系のＤＫ０，ＤＫ１に前日ファイルが収集され、インスタンスは状態αに遷移する。

【００２２】図１１は、本実施例における、図１３に示＊す物理メモリ構成を持つＮＥについての所データ変更シーケンスを示す図である。同図は図２に対応するものであり、同図において、最初の実行βの制御操作により両系のＭＭ０，ＭＭ１の中間テーブル（試験呼はこのテーブルを参照するが実呼は旧データを参照する）の所データが変更され、インスタンスは状態βに遷移する。次の実行γの制御操作により両系のＭＭの本テーブルが変更され、インスタンスが状態γ（γ１）に遷移する。実行γ２の制御操作により両系のＦＭ０，ＦＭ１が変更され、イオンスタンスはγ２に遷移する。最後の実行αにより両系のＤＫ０，ＤＫ１に前日ファイルが収集されインスタンスは状態αに遷移する。

【００２３】図１２は、本実施例における、図１４に示す物理メモリ構成を持つＮＥについてのファイル更新シーケンスを示す図である。同図は図４に対応するものであり、同図において、最初の実行βの制御操作により、１系のＭＭ１に新ファイルがロードされ、インスタンスは状態β（β１）に遷移する。次の実行β２の制御操作により、ファイル更新再開処理が行われ、ＮＥは１系で立ち上がり、インスタンスは状態β２に遷移する。最後の実行αの制御操作により、両系のＤＫ１，ＤＫ２に前日ファイル及び保証ファイルが収集され、インスタンスは状態αに遷移する。

【００２４】

【発明の効果】以上説明したように、本発明によれば、物理的なメモリ構成や更新種別に依存しないソフトウェア更新を行うための制御操作インタフェース及び制御シーケンスを実現できる。従って、ＯｐＳが物理メモリ構成や更新種別を意識することなく、ソフトウェア更新を実現することができる。即ち、制御操作インタフェース及び制御シーケンスの論理化により、ＯｐＳとＮＥの独立な発展が可能となり、相互接続性が保証される。また、制御操作インタフェース及び制御シーケンスが変更される場合でも、スーパクラスのサブクラス化により、更新種別の追加あるいは変更、新機種のＮＥの導入に対して、柔軟に対応できるので、ＮＥプログラムの変更に影響されるＯｐＳプログラムの変更は局在化され、容易に機能追加することが可能となり、ＯｐＳプログラムの流用性が向上する。

7　　8

【図面の簡単な説明】

【図１】本実施例におけるファイル更新シーケンスを示す図である。

【図２】本実施例における所データ変更シーケンスを示す図である。

【図３】本実施例におけるパッチ投入シーケンスを示す図である。

【図４】本実施例における、別の物理メモリ構成を持つＮＥのファイル更新シーケンスを示す図である。

【図５】ソフトウェア更新クラスの継承関係を示す図である。

【図６】クラスオブジェクトの継承関係を示す図である。

【図７】クラスオブジェクトの継承関係を示す図である。

【図８】クラスオブジェクトの継承関係を示す図である。

【図９】本実施例におけるファイル更新シーケンスを示す図である。

【図１０】本実施例におけるパッチ投入シーケンスを示す図である。

【図１１】本実施例における所データ変更シーケンスを示す図である。

【図１２】本実施例における、別の物理メモリ構成を持つＮＥのファイル更新シーケンスを示す図である。

【図１３】ＭＭ、ＦＭ及びＤＫの二重化メモリ構成を持つＮＥの例を示す図である。

【図１４】ＭＭ及びＤＫの二重化メモリ構成を持つＮＥの例を示す図である

【図１５】従来のファイル更新シーケンスを示す図である。

【図１６】従来のパッチ投入シーケンスを示す図である。

【図１７】従来の所データ変更を示す図である。

【図１８】従来の別の物理メモリ構成を持つＮＥのファイル更新シーケンスを示す図である。

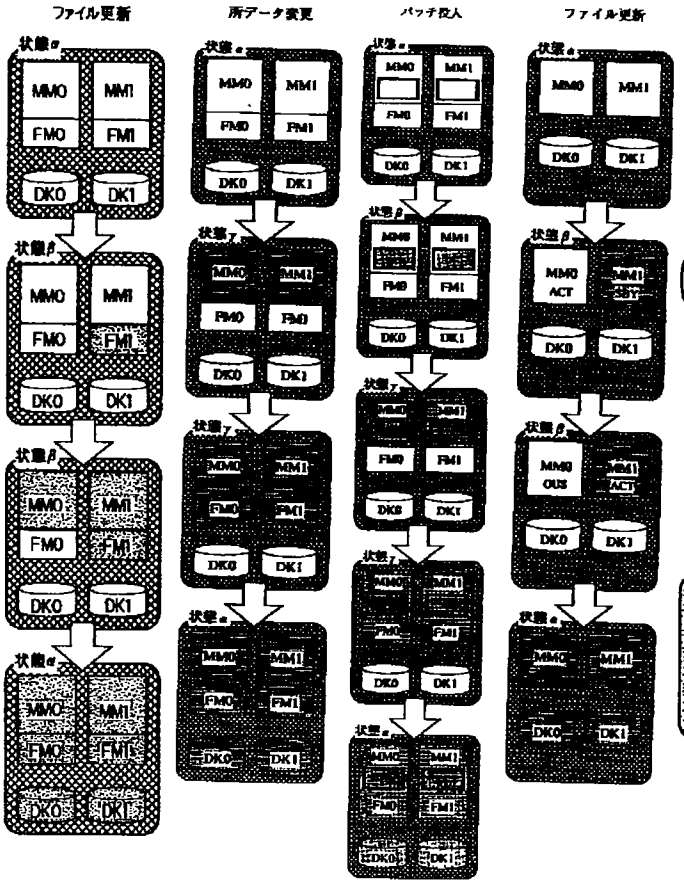
【符号の説明】

ＭＭ　メインメモリ

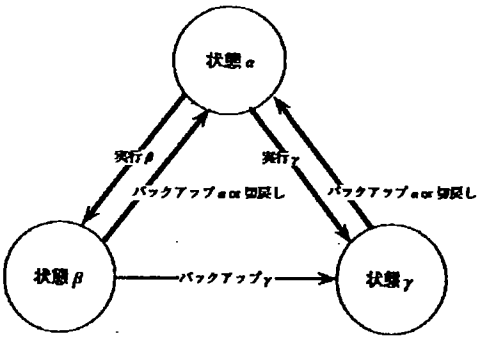
ＦＭ　ファイルメモリ

ＤＫ　ハードディスク

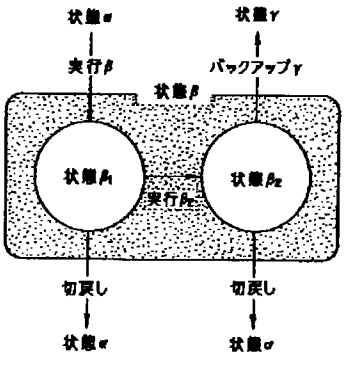
【図１】　【図２】　【図３】　【図４】



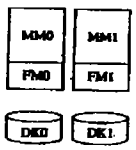
【図６】



【図７】

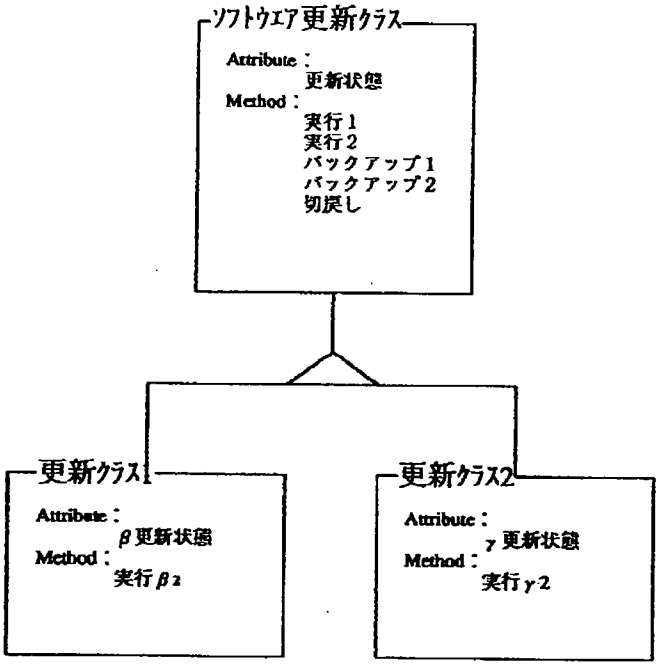


【図１３】

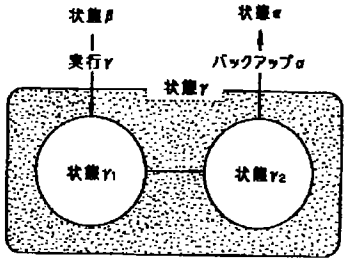


【図１４】

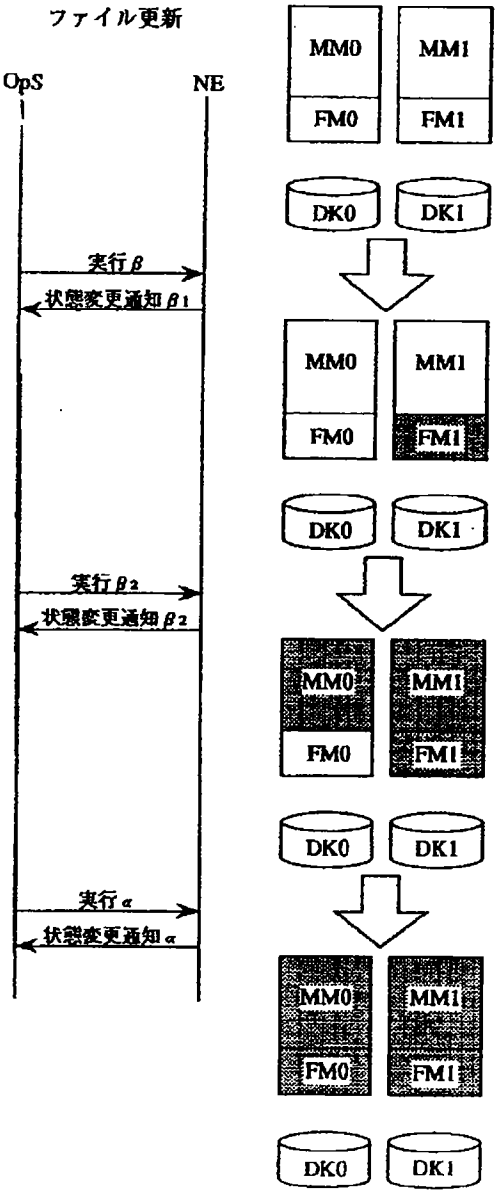
【図5】



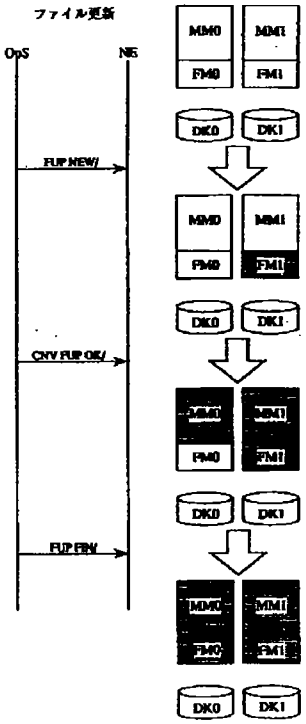
【図8】



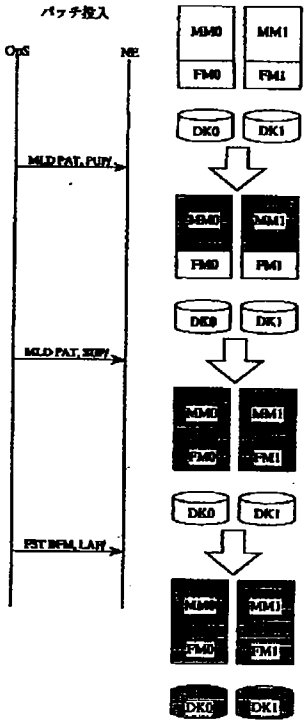
【図9】



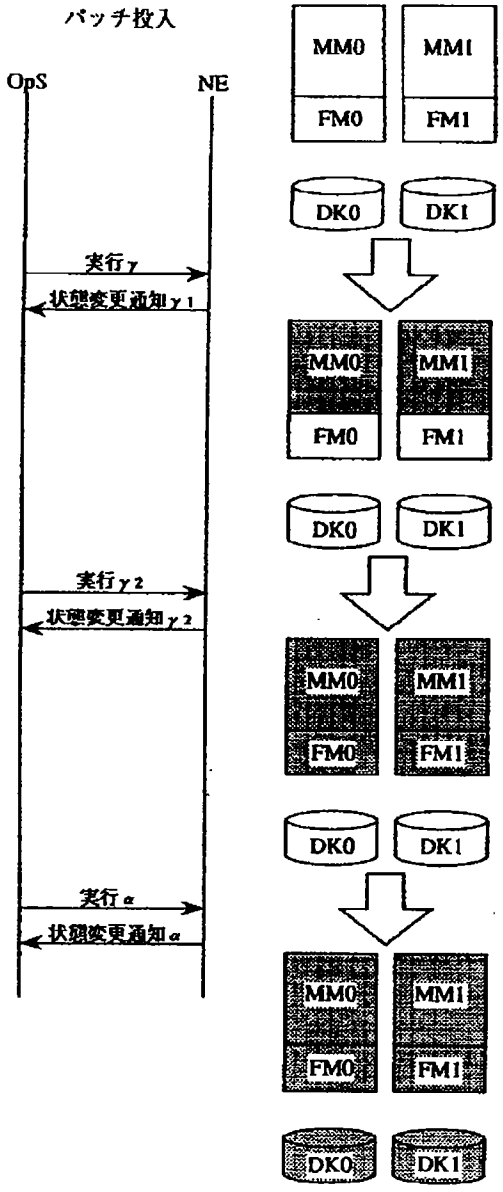
【図15】



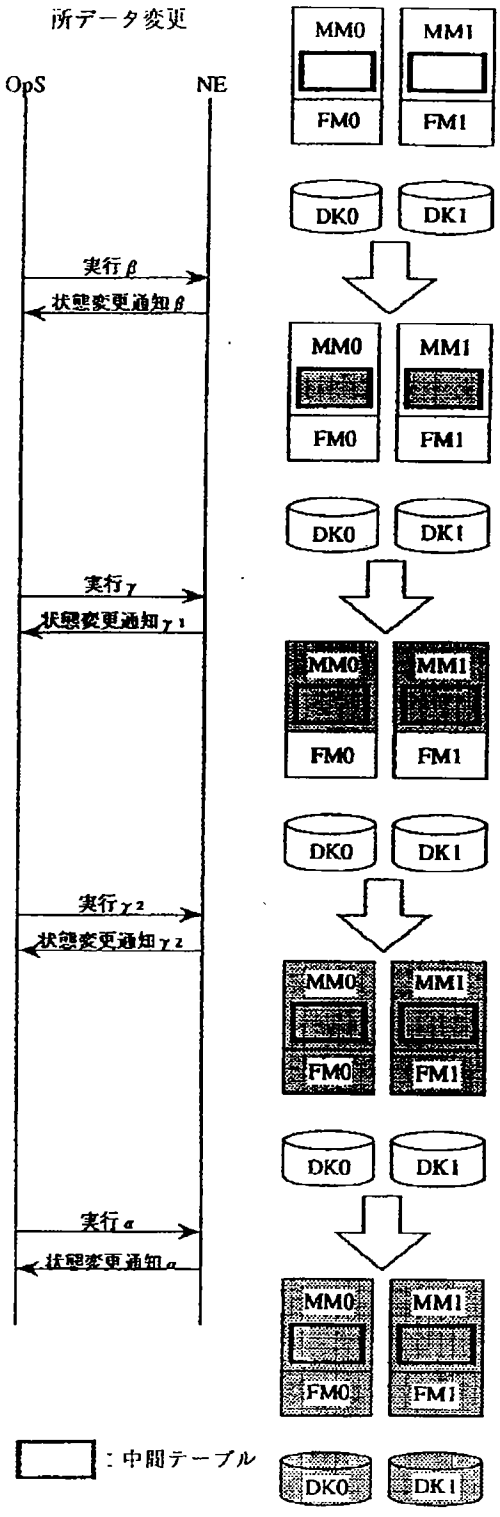
【図16】



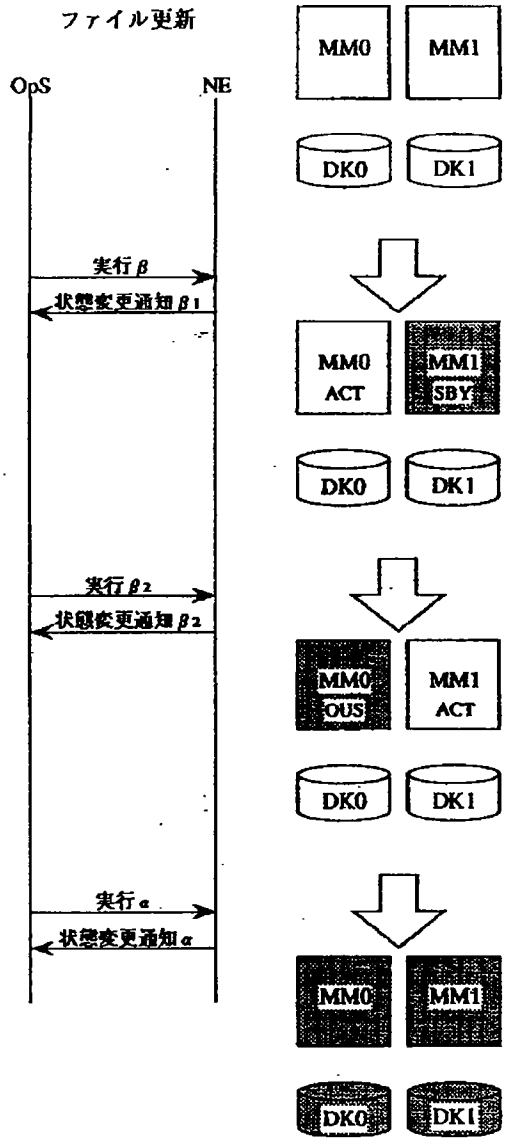
【図１０】


バッチ投入
OpS
NE
MM0
MM1
FM0
FM1
DK0
DK1
実行γ
状態変更通知γ1
実行γ2
状態変更通知γ2
実行α
状態変更通知α

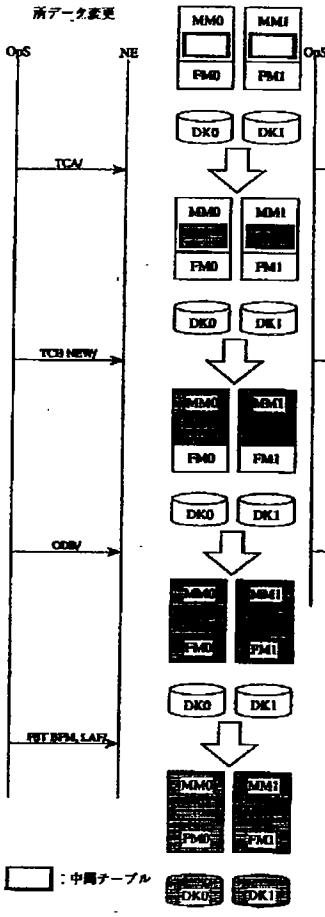
【図１１】


所データ変更
OpS
NE
MM0
MM1
FM0
FM1
DK0
DK1
実行β
状態変更通知β
実行γ
状態変更通知γ1
実行γ2
状態変更通知γ2
実行α
状態変更通知α
：中間テーブル

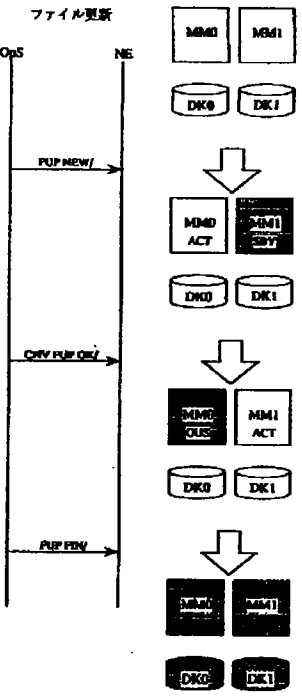
【図12】


ファイル更新
OpS
NE
実行β
状態変更通知β1
実行β2
状態変更通知β2
実行α
状態変更通知α
MM0
MM1
DK0
DK1
ACT
SBY
OUS

【図17】


OpS
NE
TCA/
TCB NEW/
ODB/
：中間テーブル

【図18】


ファイル更新
OpS
NE
PUP NEW/
PUP FIN/